Clarification of technical concepts

What are dialogue systems?

Dialogue systems are programs that "com- municate with users in natural language" (Jurafsky and Martin, 2019). These programs can be task-oriented systems, which serve to complete a task, or chatbots, which simply converse to entertain users or improve their experience with a task-oriented system (Chenet al., 2018; Jurafsky and Martin, 2019). As is the case with other language technologies,there are many approaches to building dialogue systems, depending on their function andthe resources available.

The earliest system, chatbot ELIZA, from 1966, was built using rules (Jurafsky and Martin, 2019). These rules matched word patterns in the user's input and returned outputwith a new pattern including selected elements of the input pattern and, when necessary,transformations of the input (e.g. a word like "my" in the input would become "your"in the output) (ibid). These rules also stored any input that was considered important,in order to refer back to it when new input matched relevant keywords from the memory(ibid). This architecture is still used nowadays, especially after ALICE, essentially a morecomplex version of ELIZA, was released, introducing the now very popular AIML (Artificial Intelligence Markup Language)[1] (Heller, 2016). AIML is widely employed in educational chatbots, possibly due to its ease of use (Kuyven et al., 2018).

An alternative to the time-consuming task of writing rules is using corpora to build the dialogue system. Dialogue corpora can be used to retrieve output directly, comparing the user's input to utterances in a corpus and either returning the answer to the most similar utterance, or returning the corpus response to the most similar utterance; the latter seems to yield better results (Jurafsky and Martin, 2019). Another option is using corpora to train generative, instead of retrieving responses (Jurafsky and Martin, 2019). Neural approaches offer an effective alternative and are gaining popularity (Chen et al., 2018). One advantage of neural systems is that they can return output not found in the training corpus; one disadvantage, on the other hand, is that they will normally return generic, "safe" responses (Chen et al., 2018; Jurafsky and Martin, 2019). This can be remedied by using reinforcement learning introducing variables that account for response diversity or by using adversarial networks (ibid). Though time-saving end-to-end neural systems have been developed, these require large amounts of data (Chen et al., 2018); such data may not be easily available for many languages and domains. Systems are more commonlydeveloped using a pipeline architecture, a version of the 1977 GUS architecture, which may combine rules and neural networks (Chen et al., 2018; Jurafsky and Martin, 2019). Inthese pipelines there are four key components: a Natural Language Understanding (NLU) module, a Dialogue State tracker, a set of dialogue policies, and a Natural Language Generation (NLG) module (ibid).

The NLU module processes the user's natural-language input to perform intent detection and slot filling. An intent is what the user attempts by inputting something. For instance, "I want to book a table at Giovanni's" should be assigned an intent like

[1] Artificial Intelligence Markup Language. This markup language uses pattern matching to link user input to suitable prewritten response templates. Templates can also include patterns to adapt to the context of the conversation

“request - restaurant booking”, and the completion of the task would require filling slots with entities about the time and date, number of guests and restaurant name. Slots store entitiesthat are important for the conversation, especially when a task is to be completed, andwhich slots may or must be filled depends on the intent that is assigned to the utterance (ibid). For example, continuing with the restaurant example, as the intent was “request - restaurant booking”, the system would not try to fill irrelevant slots like “song to be played” or “dress size”. The dialogue state tracker is a subsystem that analyzes the cur- rent and previous utterance, as well as the filled slots and intent; the information from the state tracker is then used to apply the right policy (ibid). In the previous example,if the dialogue state tracker observes that the intent is booking a table at a restaurantand the slot “restaurant name” has been filled directly from the request, it may select a policy where the system asks information to fill the next missing slot (e.g. date, time or number of guests), or it could also select a policy seeking confirmation if there are severalrestaurants called Giovanni’s which could fill the slot (e.g. “Giovanni’s in High Street or Giovanni’s Pizza in Clark Street?”). Finally, the NLG creates the natural-language output that matches the selected policy (ibid).

Important design principles for dialogue systems

For the design of a dialogue system, other aspects must be borne in mind besides the over-all architecture. People are used to the conventions of human dialogue, so these need tobe taken into consideration for dialogue systems to have successful conversations with people. The most popular explanation of how humans communicate successfully are Grice’sfour maxims, which can be summarized as follows: we need to give the right amount of information (enough, but not too much), we need to give true information, we need to give information that is relevant to the conversation, and give our information in a clear, organized manner (Grice, 1975, in Jurafsky & Martin, 2019). These maxims form the cooperation principle, which posits that speakers cooperate to communicate, each trying to follow the maxims for communication to be effective (ibid). Another key idea of humanconversation is that, by definition, dialogue involves more than one active participant, or else it would be a monologue (Mercer et al., 2019). A consequence of this is that participants will need to take turns to talk, and in human conversation these might overlap; spoken dialogue systems thus need to analyze prosodic cues like pauses, as well as other available cues, to detect when the user’s turn is over and the system can intervene (Jurafsky and Martin, 2019). The presence of several participants also has as consequence that theseneed to take the initiative to start and continue the conversation; in most human conversation, this role alternates between participants (ibid). Mixed initiative is very difficult for dialogue systems to achieve, which is why the initiative is normally left to the user, to avoidthe technical hurdles of mixed initiative, but also because systems constantly taking the initiative can sometimes lead to user frustration (ibid). Another important concept is the idea of implicature: people’s utterances do not always contain all the information explicitly, as they can rely on other human interlocutors to disambiguate references to previouspoints in the conversation or to shared world schemata; that is why it is important for systems to be able to keep track of the conversation and the previously mentioned entities(ibid).

Given the complexity of human dialogue, it is to be expected that there might be

miscommunications even with the most advanced systems - after all,

miscommunicationalso happens between humans. It is thus important for systems to have policies that help them deal with errors successfully (Jokinen, 2009). A sophisticated error-handling strategy distinguishes between error types (no input detected, input could not be interpreted, or system unable to process user request) to select the most appropriate response, like a reprompt (repeating the system's question), an apology, making a suggestion, etc. (Google, 2021). What response is most appropriate also depends on the system's confidence that it interpreted the input correctly and how important the information from the input mightbe: for example, before making a purchase, the system needs to be perfectly confidentthat it will perform the exact purchase that the user requested (Jurafsky and Martin, 2019). How many errors have already been encountered in the conversation is another key variable; if the system's first response to an error does not help solve it, repeating the same approach would result in an endless cycle of frustration for the user (Google, 2021).

The success of the conversation does not only depend on the system making few or noerrors and a goal being achieved; efficiency, or rather perceived efficiency is also crucial (Jokinen, 2009). How long users feel they wait for a system response has been found to be more important than actual waiting time; giving the user some form of feedback whiletheir input is further processed can help them feel that the cooperation principle is being upheld (ibid). It is also important to consider that not all tasks or task components requirenatural language understanding or generation, and they may be carried out more efficiently through a direct manipulation interface (e.g. buttons) (ibid).

Lastly, we must also mention the media equation, a theory that claims that people tend to treat computers and other media as people (Reeves & Nass, 1996, in Heller, 2011). This means that, when people talk to a dialogue system, they associate it with a certain personality (e.g. intelligent, dim-witted, submissive, funny, friendly, etc.). This is especially problematic considering that systems tend to be given female voices and names, together with stereotypical personality traits (e.g. submissiveness), which may reinforce gender roles and inequalities (Jurafsky and Martin, 2019; West et al., 2019). Therefore, dialogue system development requires making informed, careful choices about which personality the system is intended to transmit through its utterances, voice or possible avatar. Persona design also allows developers to better control that users perceive the system in a way that is more conducive to successful communication (Google, 2021), especially if this involves clearly showing the user what the system's capabilities are and how they might interact (Marge et al., 2020).

Argumentative dialogue systems

In our review of the literature on dialogic teaching, we have mentioned discussion as themain kind of interaction needed, where different participants present and argue ideas and build upon each other's contributions. Then, would dialogue systems be able to carry outthis type of talk? More research is still needed for dialogue systems to successfully participate in conversations with more than one interlocutor (Marge et al., 2020). Nonetheless, advances have been made which show that dialogue systems are capable of engaging in debates with one interlocutor. The pioneer of this type of system can be found in IBM's Project Debater (IBM, 2021). This system, given a debate topic, could scan billions of text lines and generate a four-minute opening statement defending a position; after listeningto the opponent's statement, "she" could generate a counterargument and then a closing statement summarizing the main ideas of the debate (ibid). The system is even able

to use resources that humans employ for persuasion, such as humor or appeals to emotion (ibid). Although the technical and human resources available at IBM would not be at everyone's reach, the data from the project has been made public, which could enable the creation of other debater systems. The field of argument mining provides further resources, likethe ArguAna datasets, which have been demonstrated to be usable for a rudimentarydebater system (Kulatska, 2019), or models for argument generation (Schiller et al., 2021,in Gurevych, 2021).